Simulation of a Turbulent Flow Subjected to Favorable and Adverse Pressure Gradients

Dr. Ali Uzun

National Institute of Aerospace, Hampton, Virginia

&

Dr. Mujeeb R. Malik

NASA Langley Research Center, Hampton, Virginia

Advanced Modeling & Simulation Seminar Series

20 August 2020

Introduction & Motivation

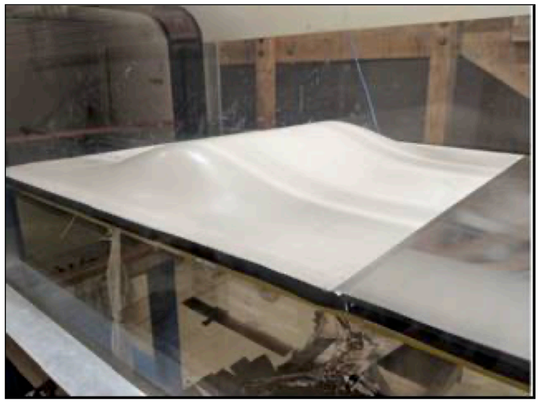
- Flows subjected to pressure gradients are hard to predict using lower-fidelity simulation tools based on various turbulence models
  - Reynolds-averaged Navier-Stokes (RANS) and wall-modeled Large-Eddy Simulation (WMLES) methods are not completely satisfactory
- A new benchmark test case called the “speed bump” has been proposed to further investigate flows subjected to pressure gradients
  - Gaussian profile that generates favorable and adverse pressure gradients
  - A detailed experimental investigation campaign is planned in near future
- We perform a direct numerical simulation (DNS) for the speed bump flow using a new flow solver developed for graphics processing units (GPUs)
- This talk reports the findings from our DNS results

3-D Speed Bump Geometry

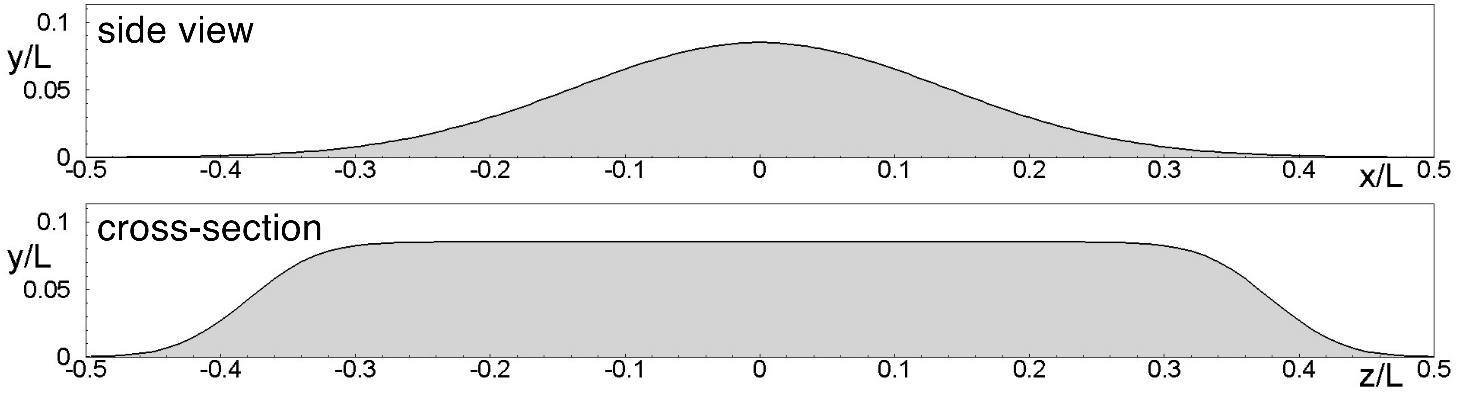
- Full geometry is defined by the following equation :

$$y(x,z) = 0.5h[1 + \mathrm{erf}((0.5\mathrm{L} - 2z_0 - |z|)/z_0)]\exp[-(x/x_0)^2]$$

where $h = 0.085L,\ x_0 = 0.195L,\ z_0 = 0.06L$



3-D geometry in tunnel



L is the span, side walls are at: $z/L = \pm 0.5$, ceiling is at: $y/L = 0.5$

Direct Numerical Simulation Methodology

- New flow solver developed exclusively for graphics processing units (GPUs)
- Compressible flow equations in generalized curvilinear coordinates
- Fully explicit schemes made up of many independent multiply-add type operations at which the GPU excels :
  - Optimized explicit 4th-order finite-difference with 6th-order filtering scheme for stability
  - 3rd-order, three-stage, explicit Runge-Kutta time integration scheme
  - 2nd-order, single-stage, Adams-Bashforth time integration scheme with improved stability
- Hybrid combination of CUDA Fortran + MPI + OpenMP :
  - CUDA Fortran to launch computational kernels on the GPU
  - Host-assisted MPI communication among GPUs (“GPU-aware MPI” performance is very poor)
  - OpenMP for running multiple threads on the host CPU
    - Master thread controls the GPU and launches the kernels
    - 6 helper threads handle the MPI communication (one thread per face of a mesh block)

GPU Code Performance

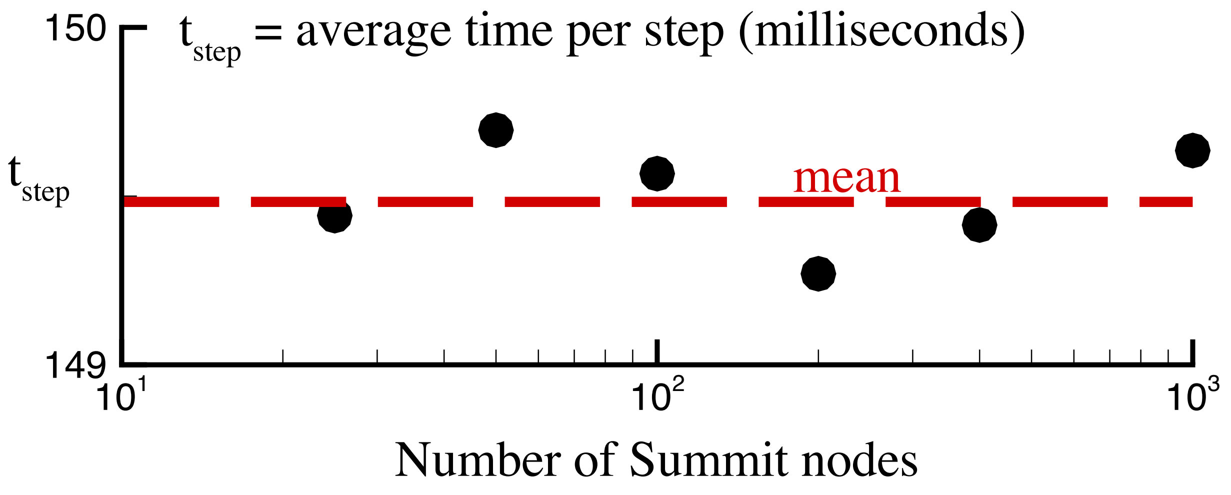
- GPU code provides a ~7.5x speedup over our CPU code :
  - Comparison between one Intel Skylake node with 40 cores and one NVIDIA Tesla V100 GPU with 16 Gigabytes of global memory
  - These hardware are similar in price and power consumption
  - Based on the shortest time taken to run a problem over a given physical time interval
- CPU code uses :
  - High-order compact finite-difference schemes (which require a scalar tridiagonal solver)
  - Explicit (third-order Runge-Kutta) or implicit (Beam-Warming) time integration schemes
  - Same explicit scheme also used in the GPU code
  - Max CFL number of about 1.3 with the explicit time integration scheme
  - Max CFL number of about 5 for time accuracy with the implicit time integration scheme
- CPU code is fairly-well optimized for Intel CPUs

GPU Code Parallel Scaling

- MPI communication is very slow compared to GPU computation :
  - If assigned too little work, the GPUs would sit idle while waiting for MPI to catch up
  - Poor strong-scaling performance because of slow communication relative to computation
  - To make the best use of limited resources, we assign the maximum possible workload to the GPU and overlap as much communication with computation
  - In other words, we try to minimize the total node-hours, rather than the wall-clock run time
- We assign max workload to the GPU by maxing out the available memory :
  - Nearly 50 million points per 32 Gigabytes of memory on a NVIDIA Tesla V100 GPU
  - All data arrays are stored in the GPU memory
  - Chosen explicit schemes allow decoupling of points near block interfaces from interior points
  - Host-assisted MPI communication is overlapped with GPU computation
  - Multiple OpenMP threads running on the host CPU handle the MPI communication
  - Overlapping strategy leads to complicated code structure

Weak-Scaling Performance on Summit

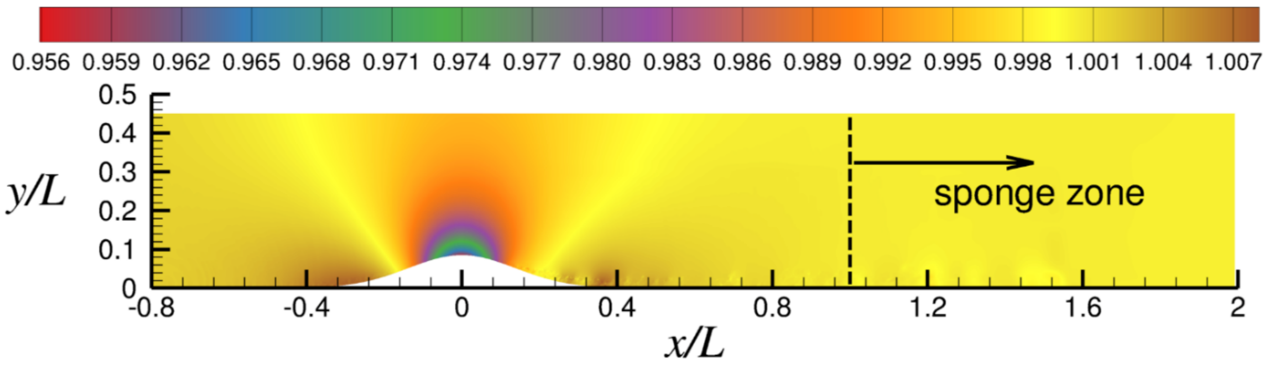
- Test runs were performed with 142 million points per Summit node
  - Summit has 6 V100 GPUs per node, each with 16 Gigabytes of memory
- Number of nodes is increased while keeping the number of points per node fixed



- About one nanosecond per grid point per Summit node per time time step

Spanwise-Periodic DNS Schematic

- Limited resources constrain us to spanwise-periodic DNS, with span set to 0.04L
- Reynolds number (based on upstream velocity and L) = 1 million
  - Inflow momentum-thickness Reynolds number, $Re_\theta \approx 530$; at $x/L = -0.4,\ Re_\theta \approx 1600$
- Freestream Mach number = 0.2 (close to the expected experimental value)
- RANS data to set up mean inflow and top plane boundary conditions
- Turbulent inflow generation based on the recycling/rescaling method
- Viscous isothermal boundary condition on lower wall
- Sponge zone with characteristic boundary condition on outflow boundary



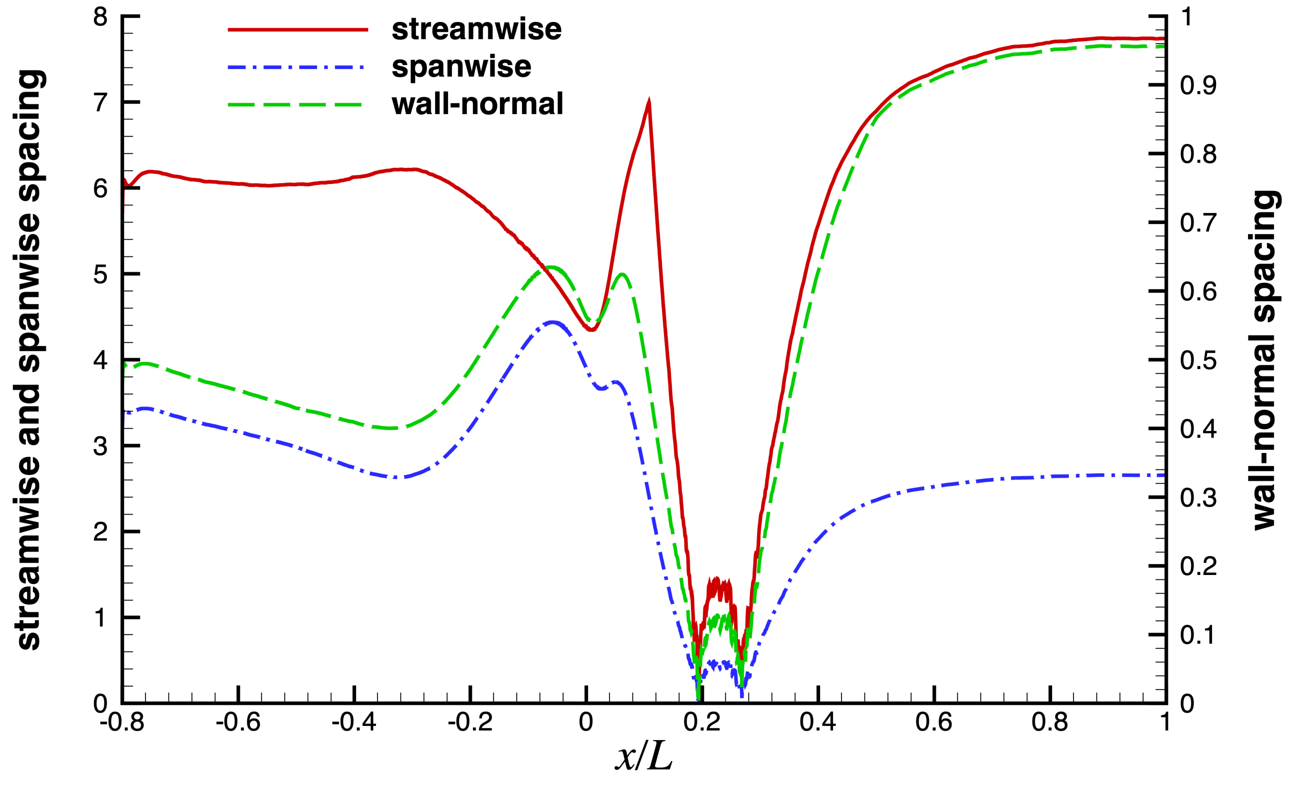
Contours denote the instantaneous pressure normalized by reference value

Direct Numerical Simulation Details

- 2.265 billion grid points total
- Grid resolutions in wall units are discussed on the next slide
- Optimized explicit 4th-order finite-difference with 6th-order filtering scheme
- 3rd-order, three-stage, explicit Runge-Kutta time integration scheme
- 750,000 steps to compute a time interval of L/U_∞ (max CFL number of 0.86)
- 48 NVIDIA V100 GPUs at NAS, each with 32 GB memory, to perform the DNS
- Two months of run time to compute a time interval of $20\,L/U_\infty$
- Code is stable up to the CFL number of about 1.3
- At CFL = 1.3, the DNS would have taken about 40 days
- Flow statistics are gathered over $15\,L/U_\infty$

NATIONAL INSTITUTE OF AEROSPACE

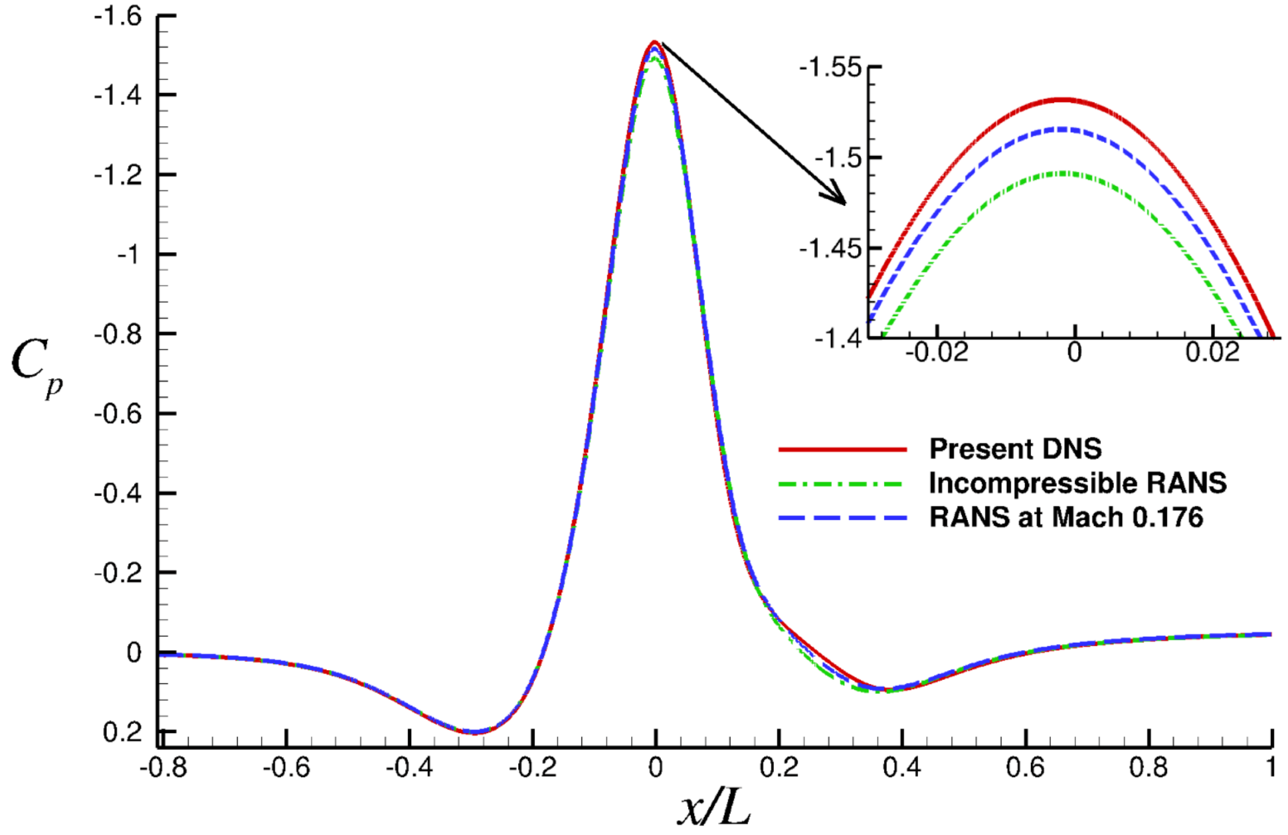
Streamwise Variation of Grid Resolutions



- Largest streamwise spacing in wall units: 6 to 8 units
- Largest spanwise spacing in wall units: 4.5 units
- Wall-normal spacing at wall: 0.4 to 0.95 units
- Largest wall-normal spacing around boundary layer edge: about 10 units

NATIONAL INSTITUTE OF AEROSPACE

Surface Pressure Coefficient Distribution & Comparison with RANS



RANS calculations with the Spalart-Allmaras model were performed by colleagues in separate independent studies

The sense of streamwise pressure gradient changes at the foot, apex and tail of the bump; this will trigger the formation of internal layers

Determination of Mean Boundary Layer Properties

- Spalart and Watmuff (*JFM*, 1993) proposed the following integrals based on spanwise vorticity to compute boundary layer edge velocity & displacement and momentum thicknesses for incompressible flow

$$U_e = \int_0^{\infty} -\omega_z(y_w)dy_w$$

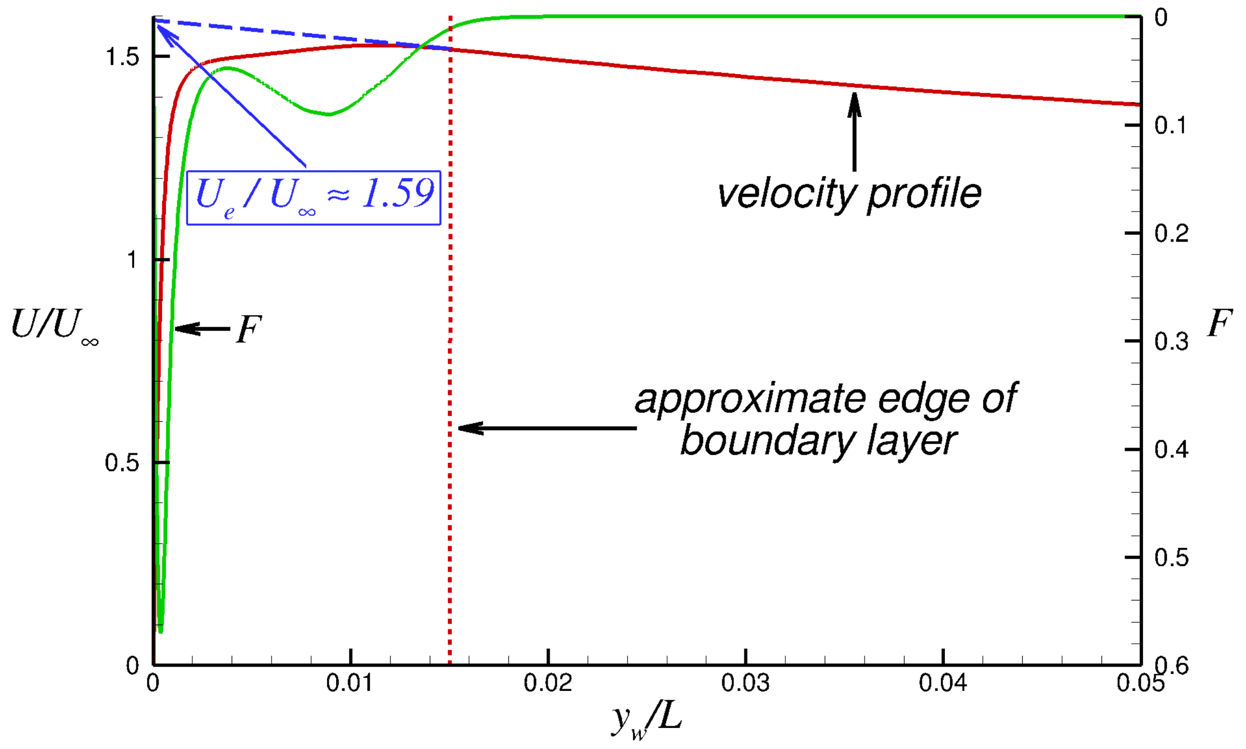
$$\delta^* = \frac{1}{U_e}\int_0^{\infty} -y_w\omega_z(y_w)dy_w$$

$$\theta = \frac{2}{U_e^2}\int_0^{\infty} y_w\left[\int_0^{y_w} \omega_z(n)dn\right]\omega_z(y_w)dy_w - \delta^*$$

y_w is the wall-normal distance, ω_z is the mean spanwise vorticity

NATIONAL INSTITUTE OF AEROSPACE

Determination of Mean Boundary Layer Thickness



Velocity and F profiles at the apex are shown

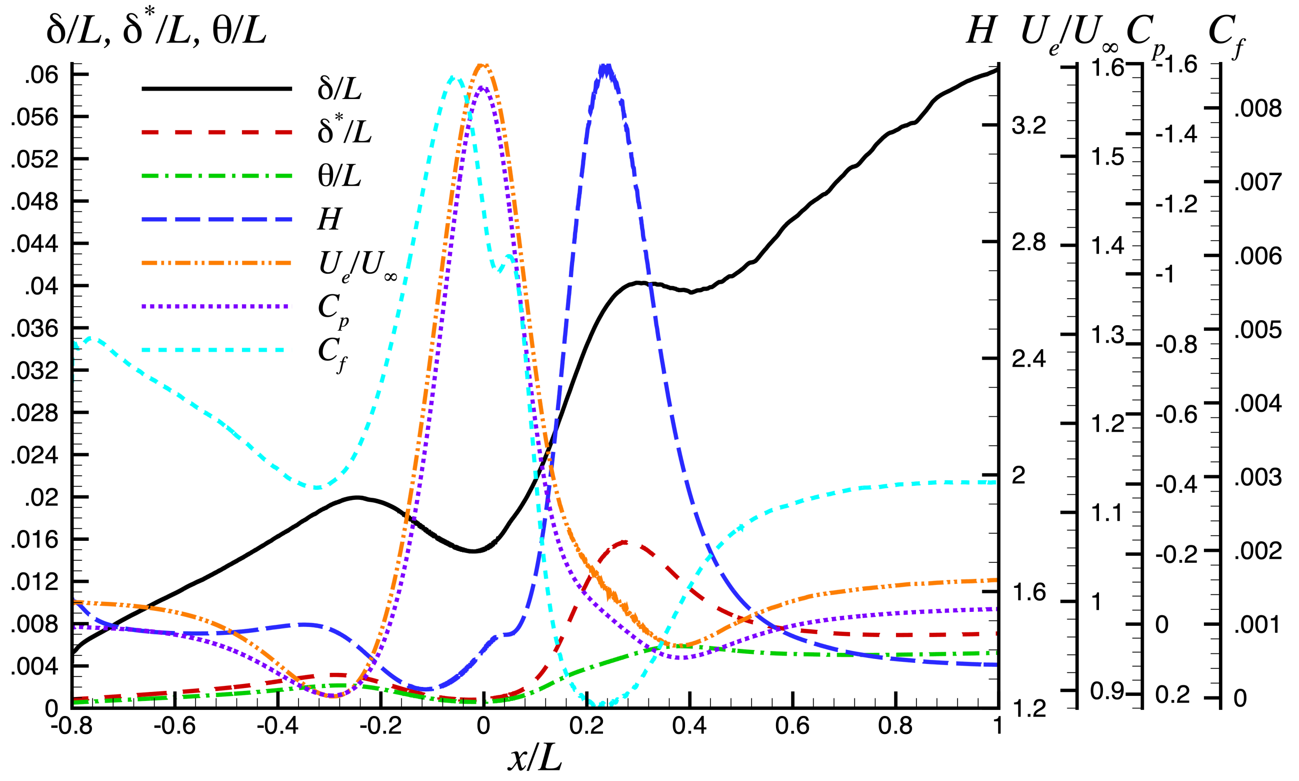
$$F = -y_w \omega_z$$
at $y_w/L \approx 0.015$
$$F \approx 0.02 F_{max}$$

Extrapolation of velocity at $y_w/L \approx 0.015$ to wall gives an estimate of U_e that agrees with the value from vorticity integral (credit is due Dr. Spalart)

NATIONAL INSTITUTE OF AEROSPACE

Variation of Mean Boundary Layer Properties

- Flow goes through significant changes caused by pressure gradients

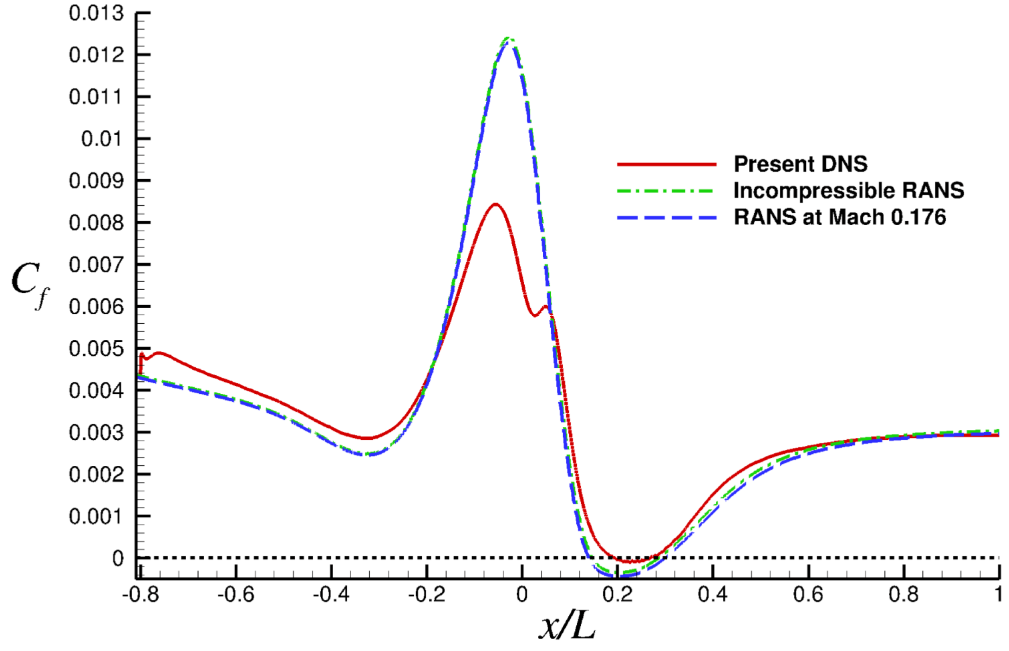


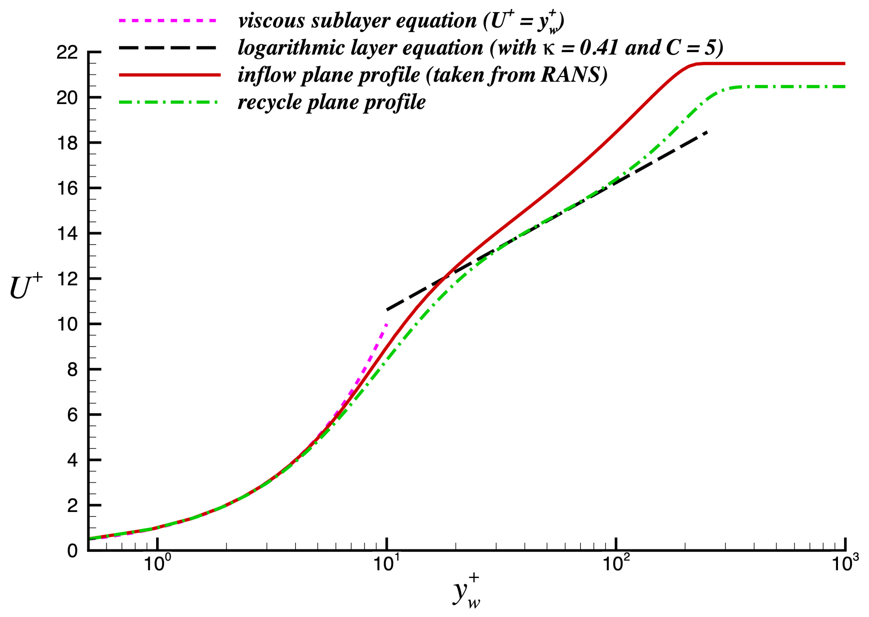
From left to right, the quantities plotted:

- boundary layer thickness
- displacement thickness
- momentum thickness
- shape factor
- edge velocity
- surface pressure coefficient
- skin-friction coefficient

National Institute of Aerospace

Skin-Friction Coefficient Distribution & Comparison with RANS





RANS calculations predict much higher peak skin friction levels and more severe separation

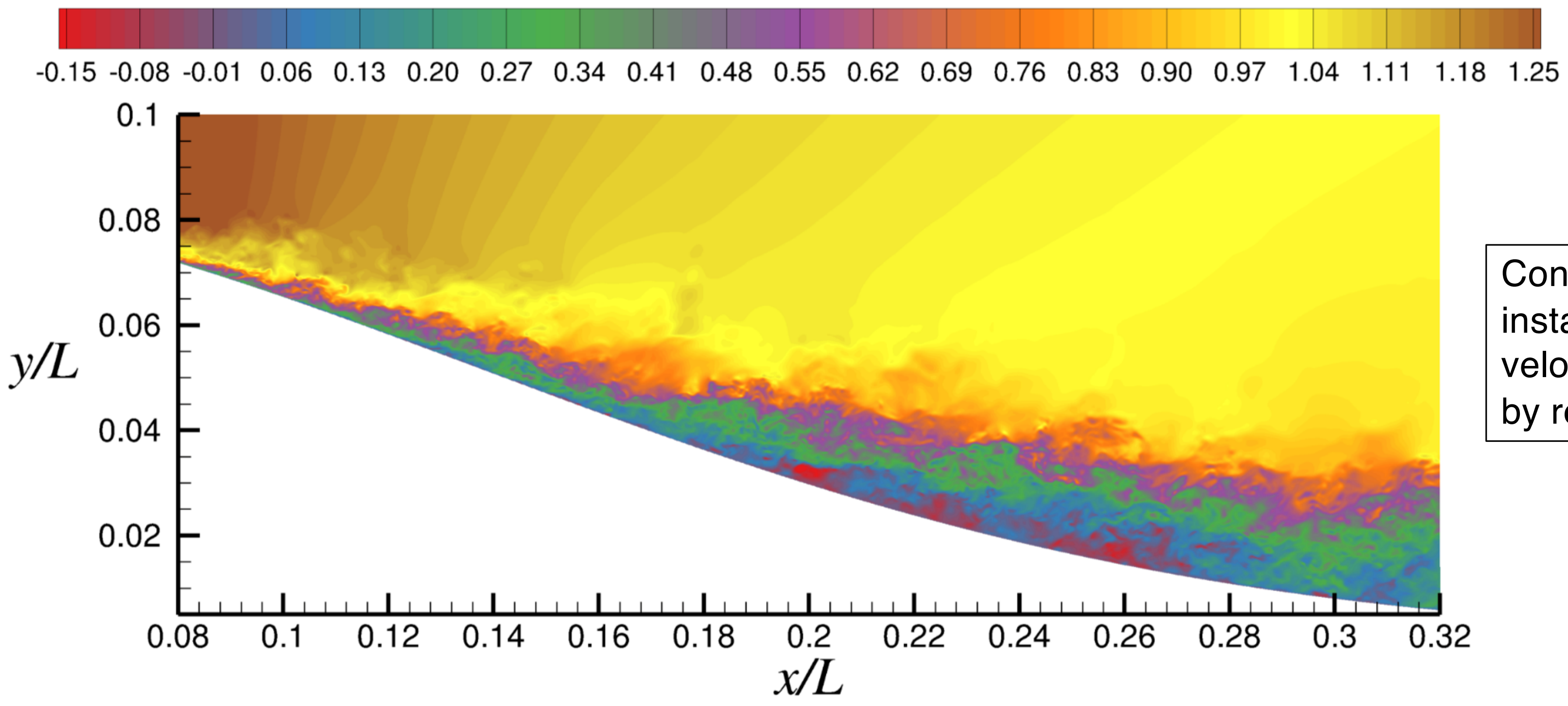
As expected, RANS does not detect relaminarization

The jump in the DNS C_f near the inflow is due to erroneous mean inflow profile taken from RANS

NATIONAL INSTITUTE OF AEROSPACE

Incipient or Very Weak Separation in Decelerating Region

- Flow visualization shows incipient or very weak separation in the aft section

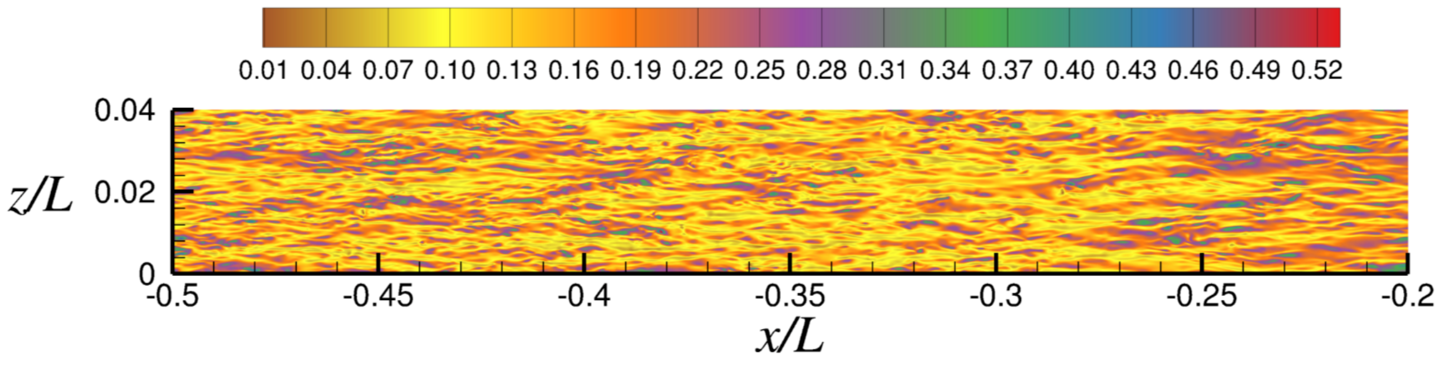


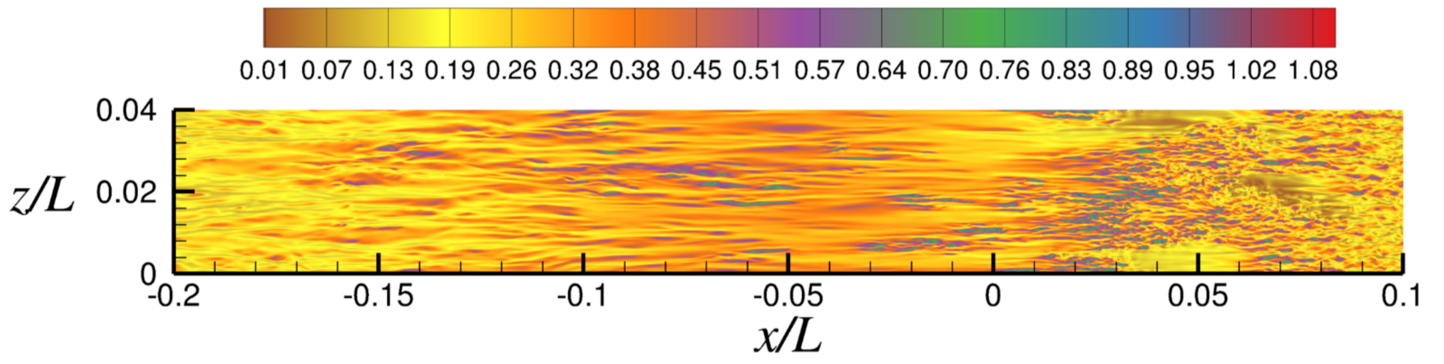
Contours denote the instantaneous axial velocity normalized by reference velocity

NATIONAL INSTITUTE OF AEROSPACE

Flow Relaminarization/Stabilization in Accelerating Region

- Combination of low Reynolds number, strong favorable pressure gradient and convex curvature leads to flow relaminarization/stabilization upstream of apex



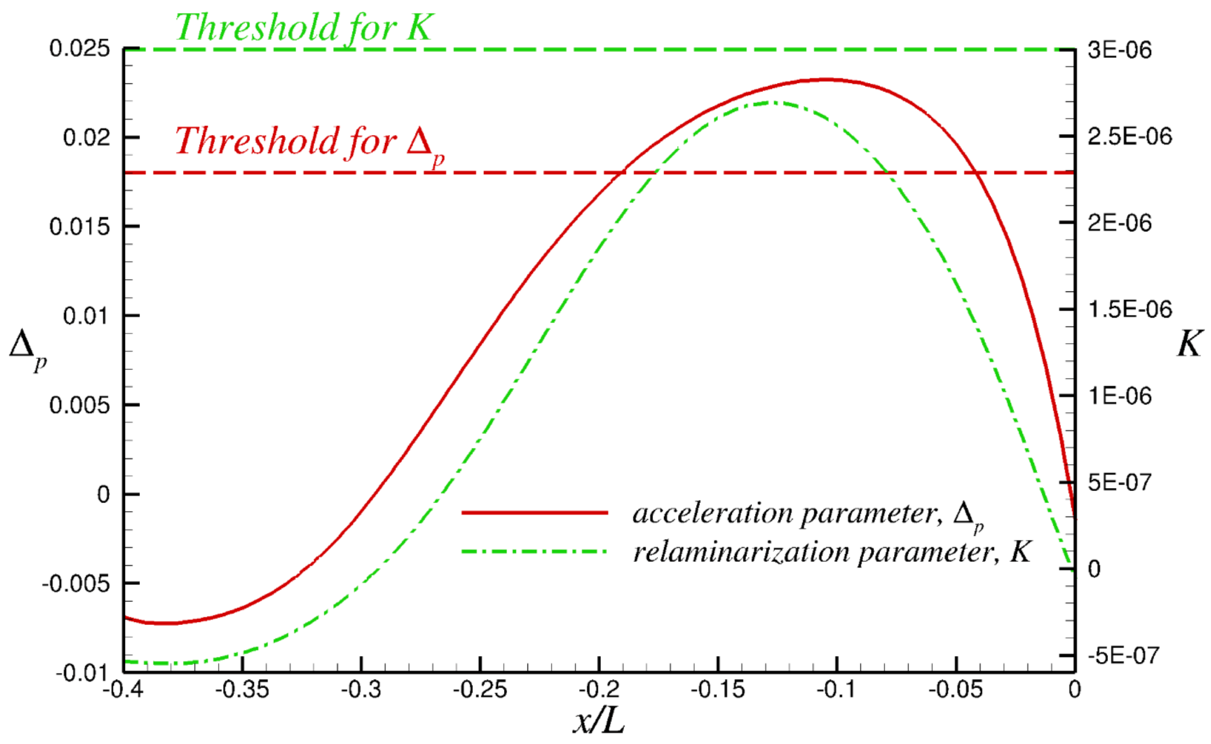


Normalized total velocity contours on a near-wall plane where $4 \lesssim y_w^+ \lesssim 6$

y_w^+ is the wall-normal distance in wall units

Variation of Acceleration and Relaminarization Parameters

- Variation of acceleration and relaminarization parameters (Δ_p and K):



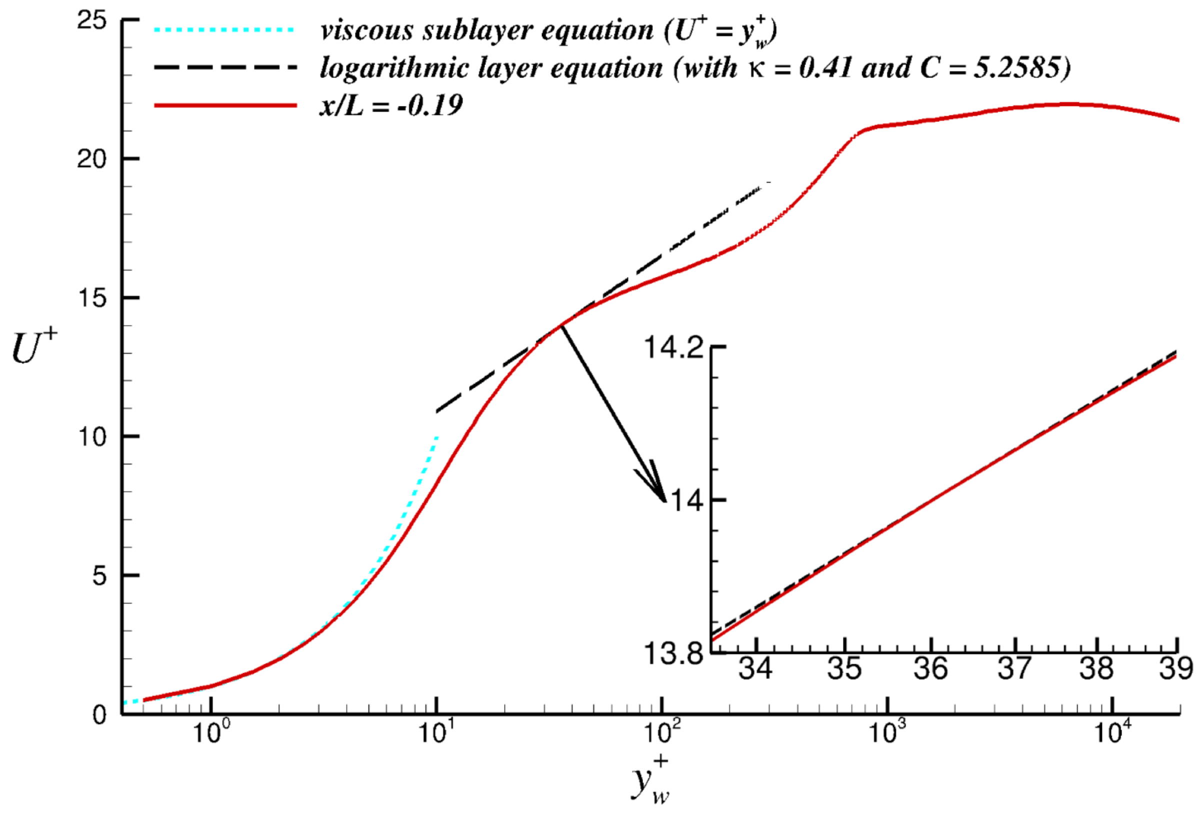
$$\Delta_p = -\frac{\nu}{\rho u_\tau^3}\frac{\partial p}{\partial s}$$

$$K = \frac{\nu}{U_e^2}\frac{\partial U_e}{\partial s}$$

ρ is the density
ν is the kinematic viscosity
u_τ is the friction velocity
p is the pressure
s is the surface distance
U_e is the edge velocity

National Institute of Aerospace

Logarithmic Layer Breakdown in Accelerating Region



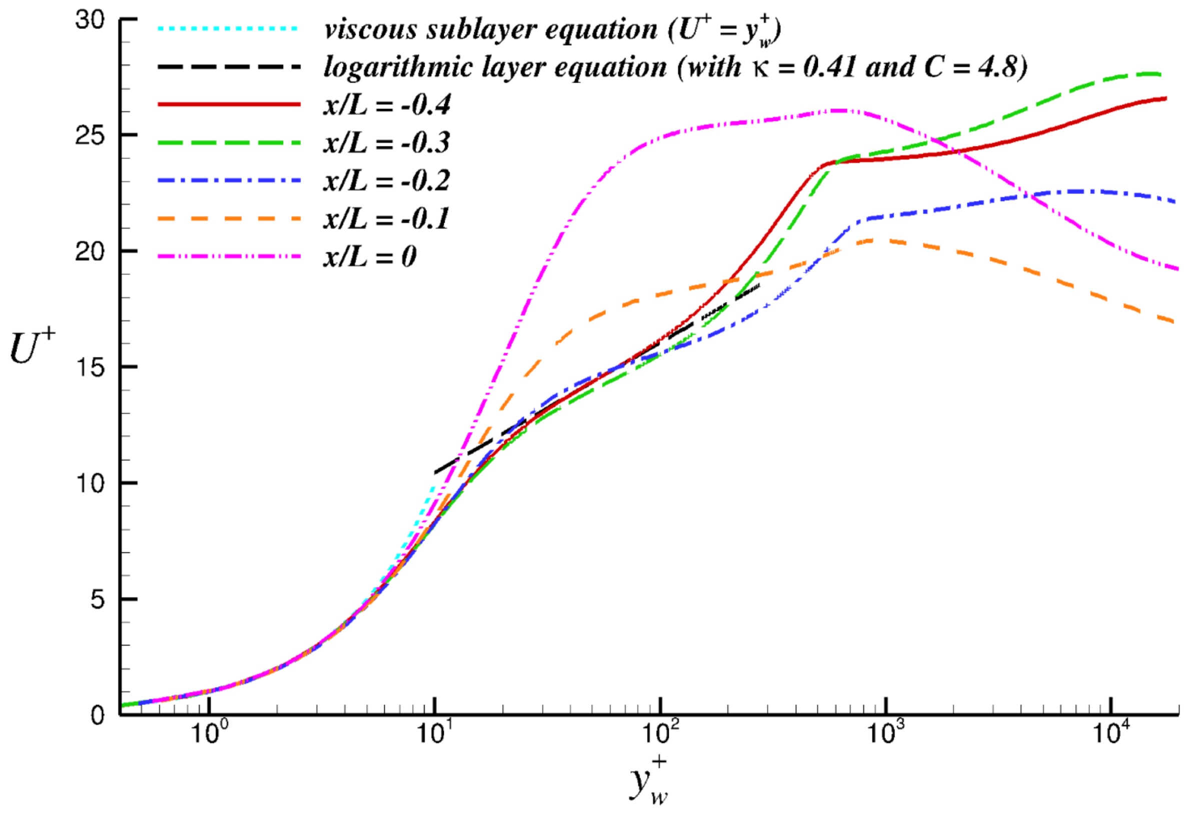
$\Delta_p = 0.018$ at $\frac{x}{L} \approx -0.19$

Patel and Head (*JFM*, 1968) concluded that “major departures” from logarithmic layer occur when Δ_p exceeds the threshold value of 0.018

The logarithmic layer has nearly disappeared where $\Delta_p = 0.018$

NATIONAL INSTITUTE OF AEROSPACE

Mean Velocity Profiles in Accelerating Region



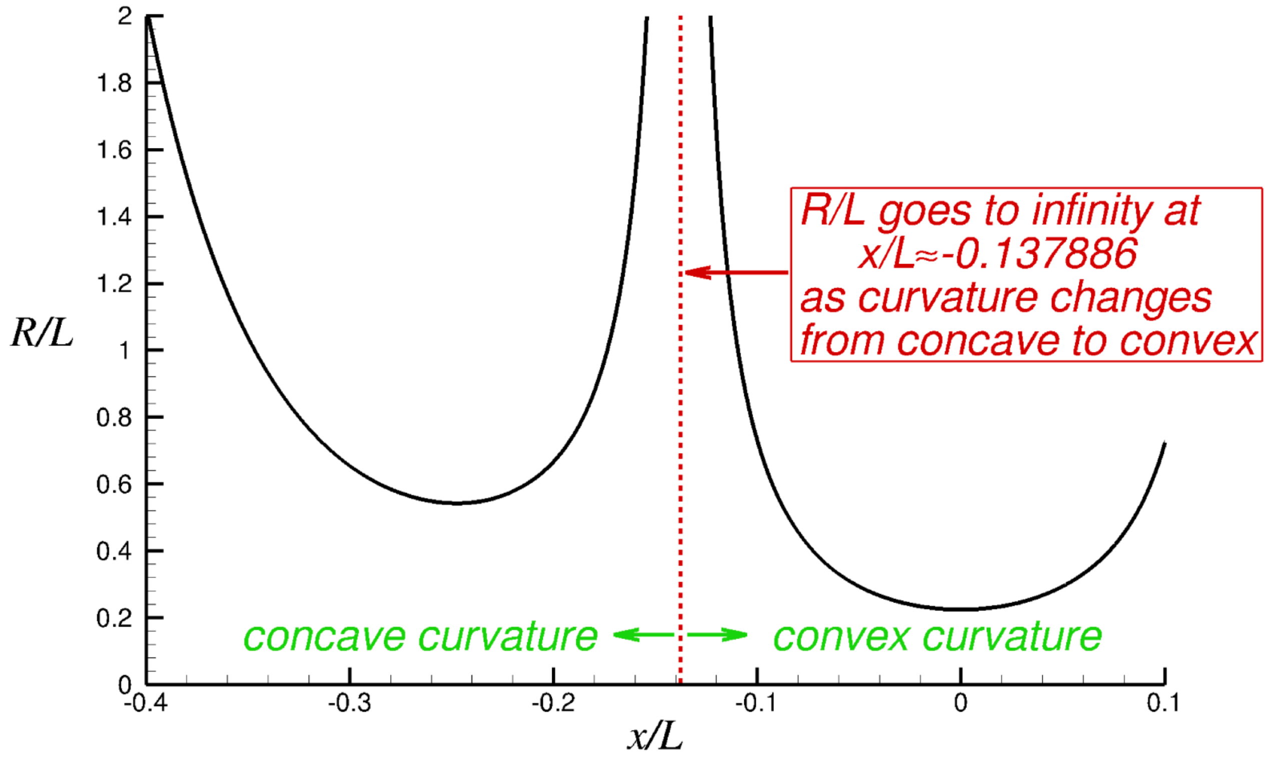
$$y_w^+ = y_w u_\tau / \nu$$
$$U^+ = U / u_\tau$$

ν is the kinematic viscosity
u_τ is the friction velocity
U is the streamwise velocity
y_w is the wall distance

Velocity profiles take shapes resembling those of relaminarizing boundary layers as the accelerated flow approaches the apex

NATIONAL INSTITUTE OF AEROSPACE

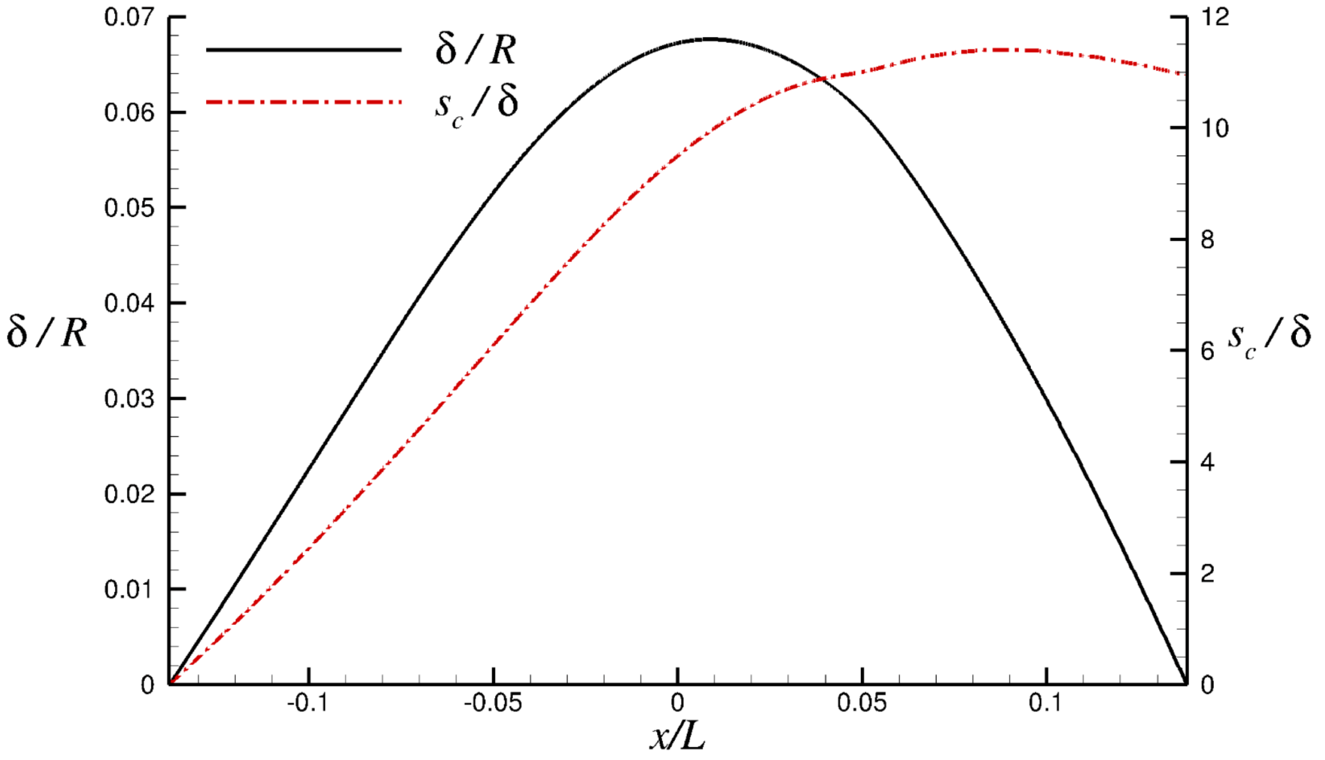
Surface Curvature Variation



Because of the geometrical symmetry with respect to $\frac{x}{L} = 0$, surface curvature variation is also symmetric with respect to $\frac{x}{L} = 0$

NATIONAL INSTITUTE OF AEROSPACE

Convex Surface Curvature Effect



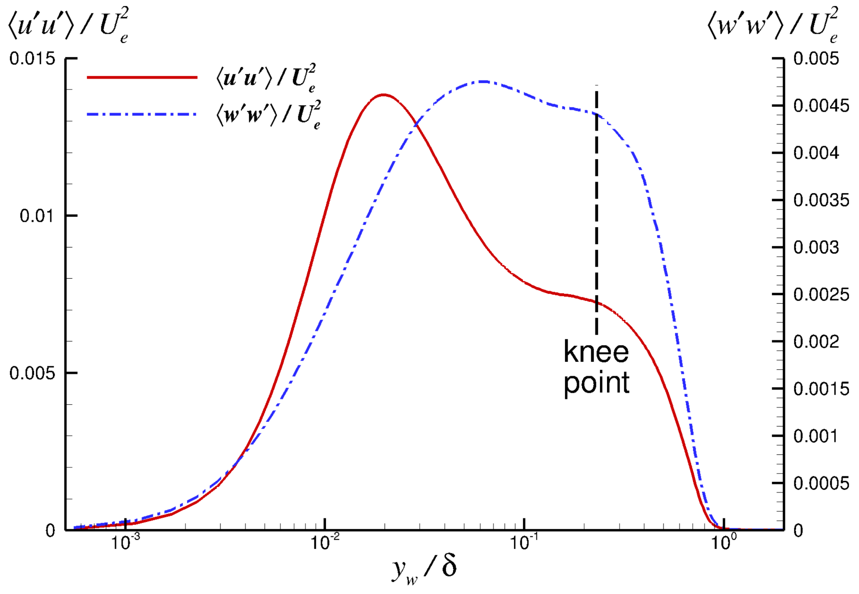
s_c is the surface distance measured from the start of convex curvature

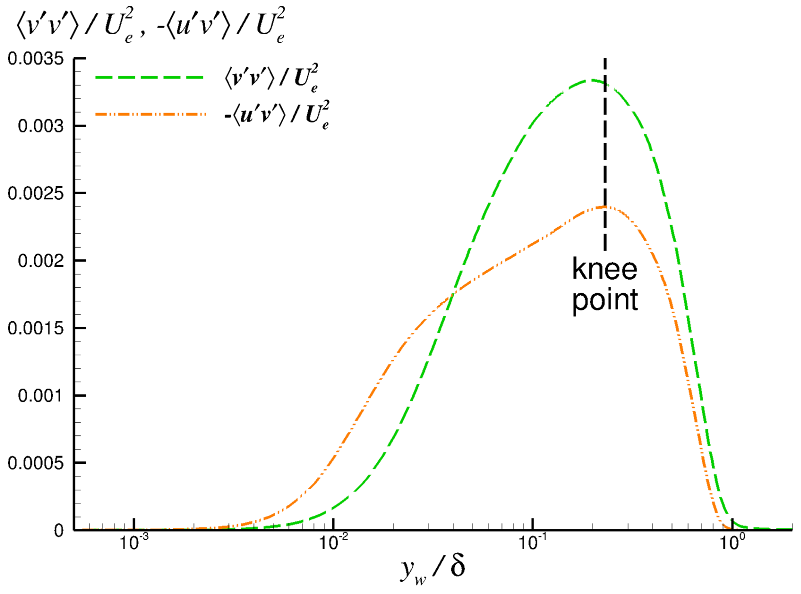
δ / R significantly exceeds "mild" convex curvature value of ~0.01

Reynolds Stress Profiles at $x/L = -0.3$

Emergence of a “knee point” at $y_w/\delta \approx 0.23$ in the streamwise and spanwise stresses

Alignment of the “knee point” relative to the peaks in the wall-normal and shear stresses

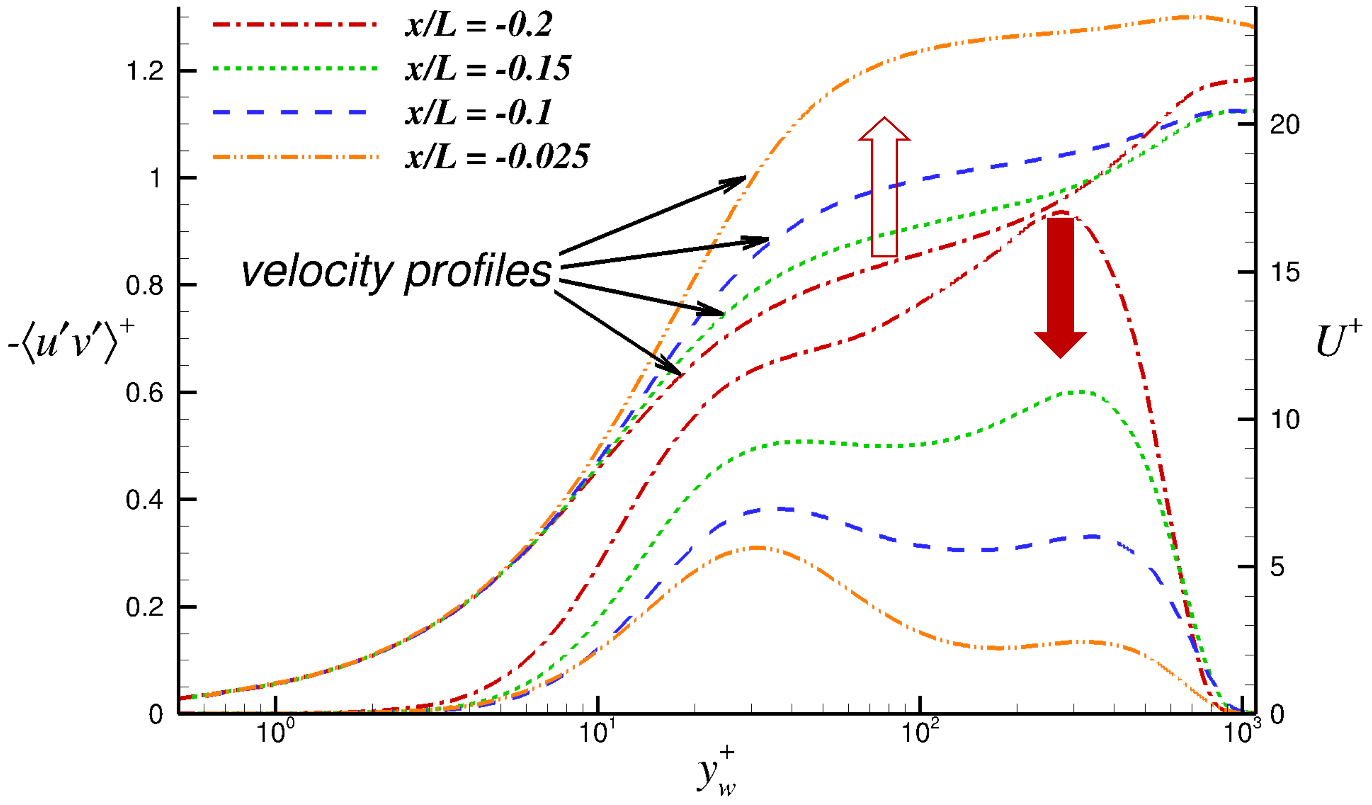




The “knee point” emerging in the streamwise & spanwise stresses indicates an internal layer triggered by the change from adverse to favorable pressure gradient at bump foot

NATIONAL INSTITUTE OF AEROSPACE

Shear Stress and Velocity Profiles in Accelerating Region



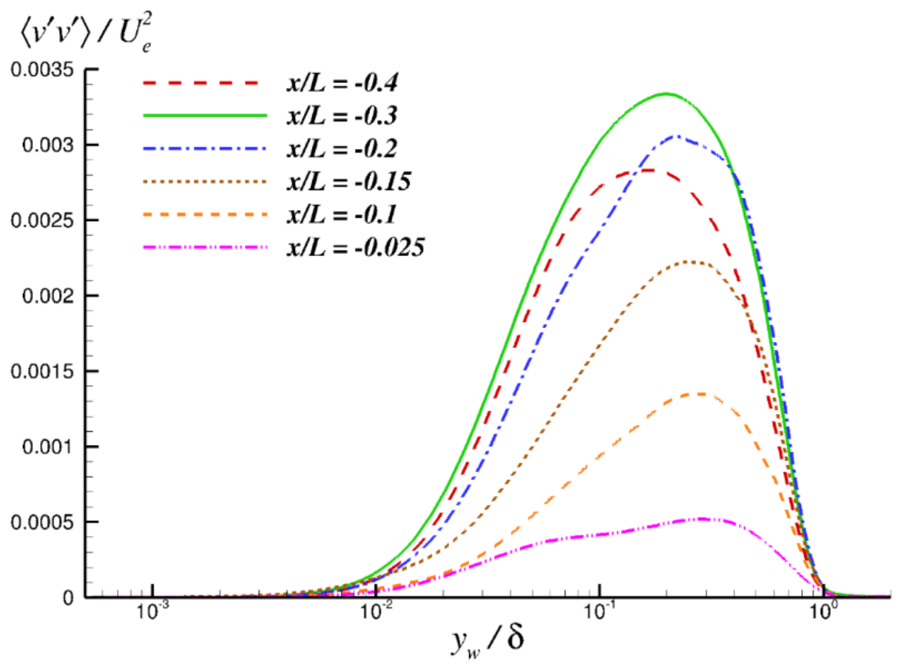
An inner stress peak develops in the internal layer, but its growth is hampered by flow stabilization

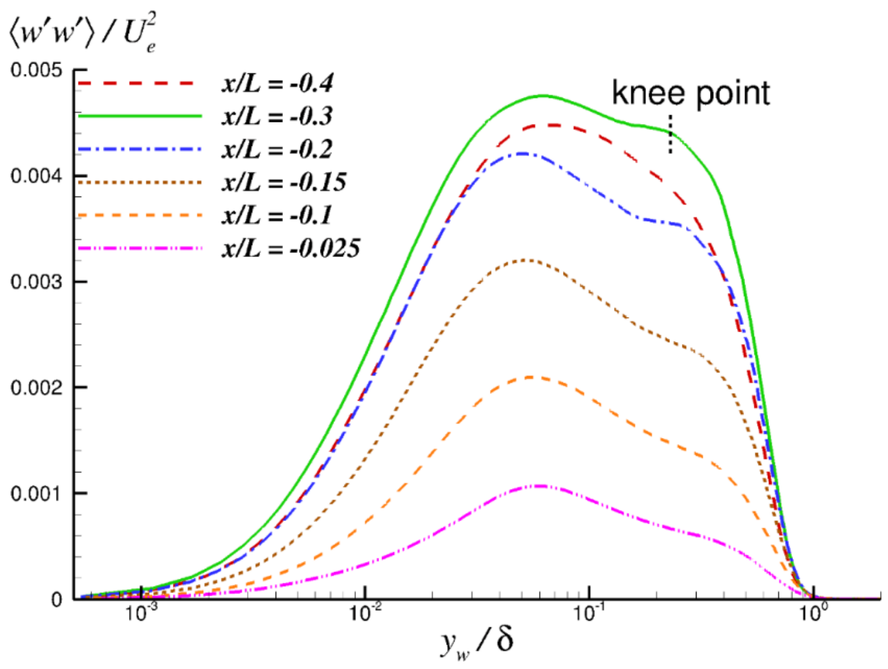
Original outer peak decays due to reduction in velocity gradient in outer region by flow acceleration

NATIONAL INSTITUTE OF AEROSPACE

Mixing Suppression in Accelerating Region

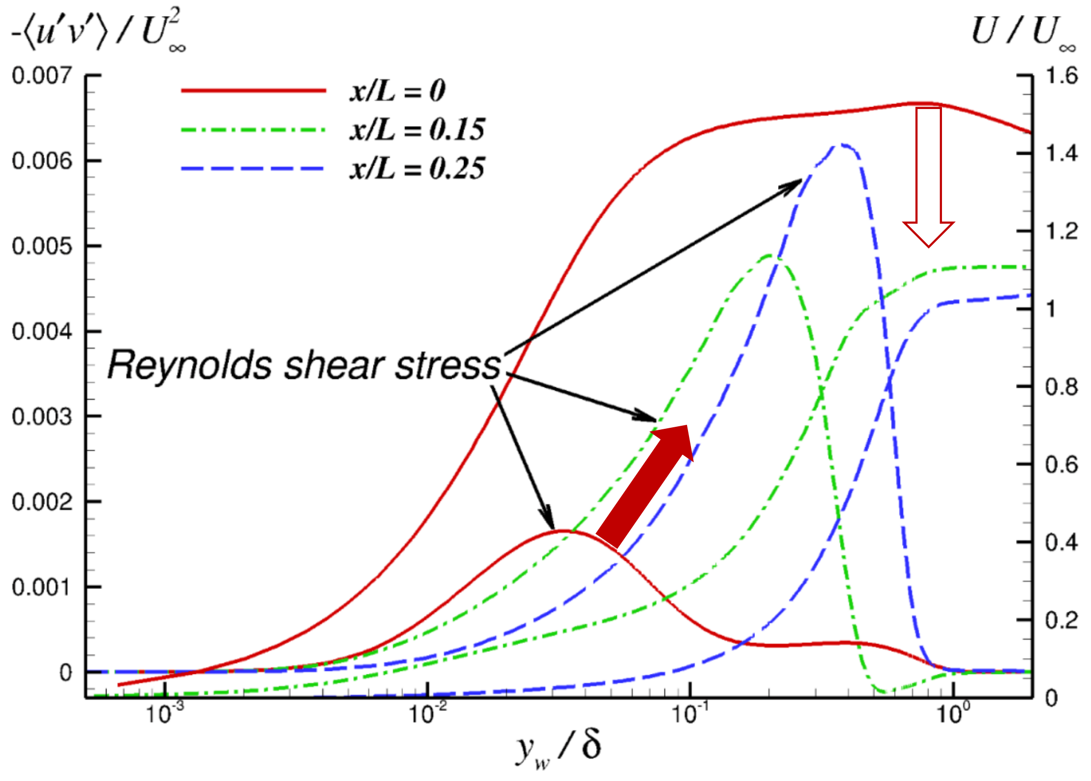
- Significant reduction in wall-normal and spanwise Reynolds stresses





National Institute of Aerospace

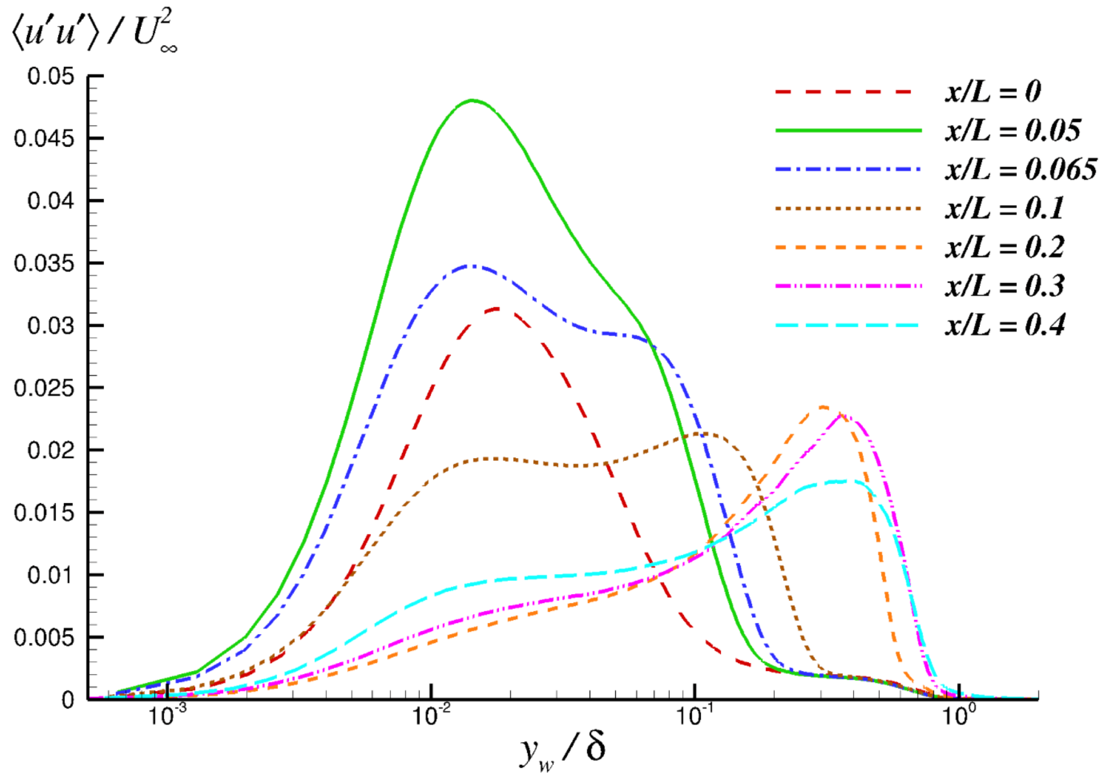
Shear Stress and Velocity Profiles in Decelerating Region



Deceleration leads to the formation of a buffer layer between low- and high-speed regions; this buffer zone acts like a free shear layer and can be viewed as a new internal layer

Shear stress peak shifts from the near-wall region to the free shear layer, which also contains the normal stress peaks

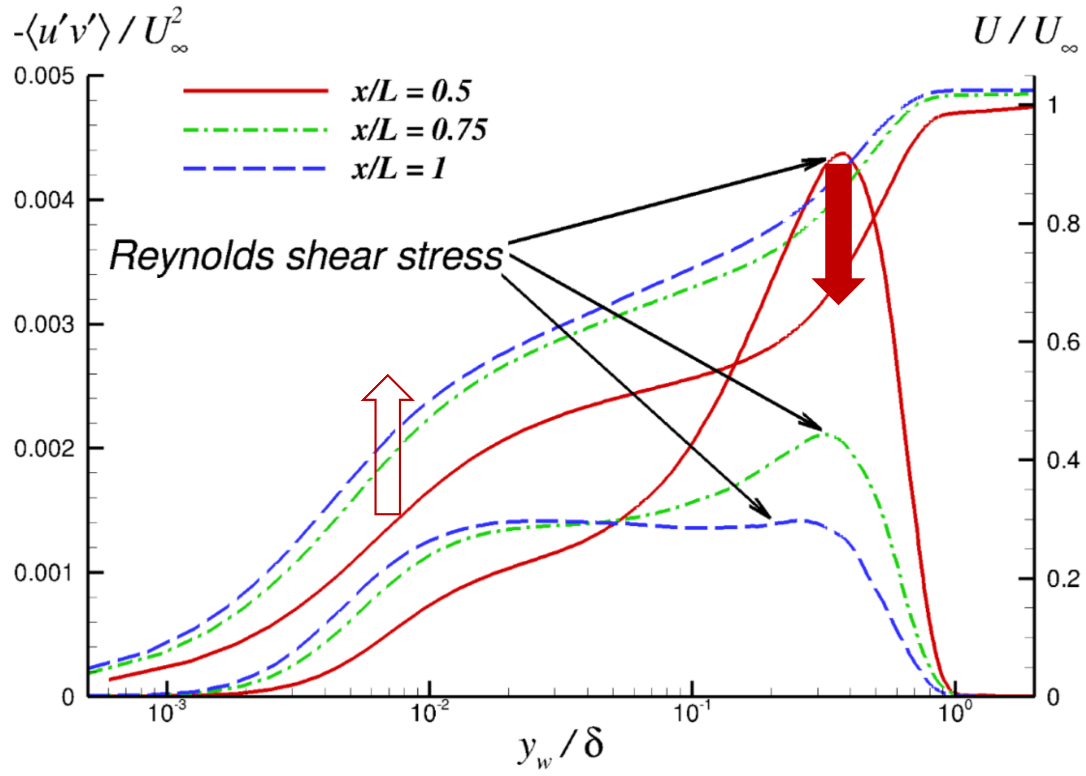
Streamwise Reynolds Stresses in Decelerating Region



Inner peak strengthens first due to the retransition to turbulence but weakens as the flow decelerates; an outer peak emerges in the free shear layer region

NATIONAL INSTITUTE OF AEROSPACE

Shear Stress and Velocity Profiles in Recovery Region

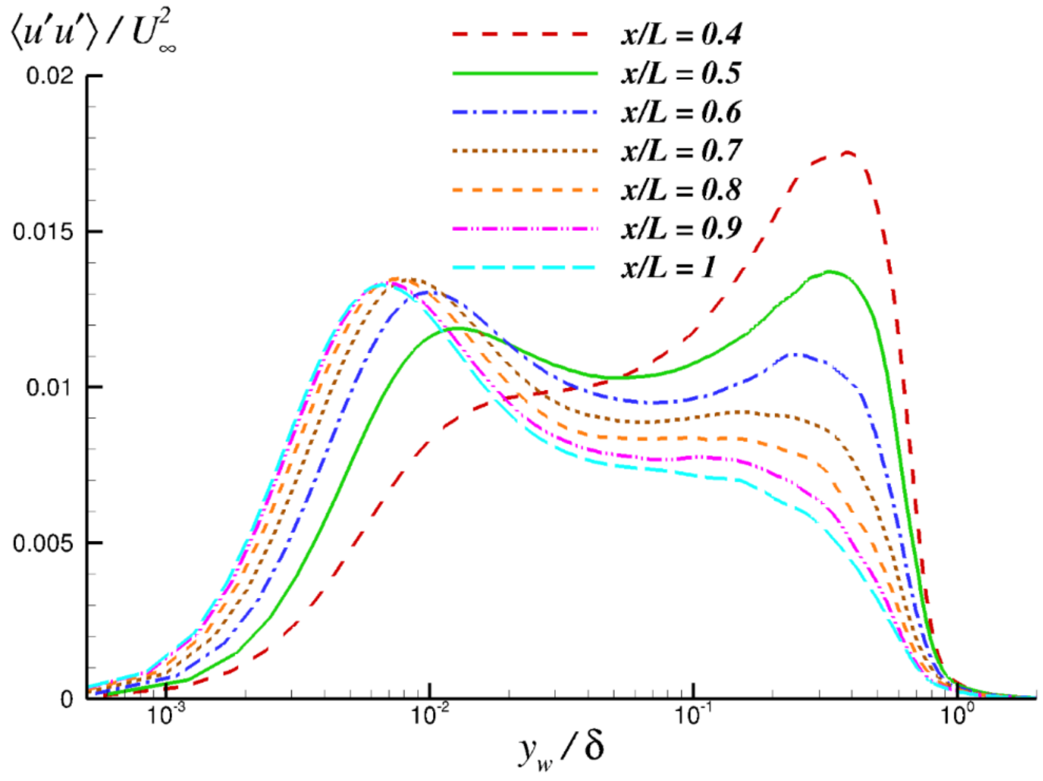


The weakly favorable pressure gradient in recovery region reduces the velocity gradient in the free shear layer region; shear and normal stress peaks start to decay

Another internal layer develops near the wall; but the domain is not long enough for the inner stress peak to fully form

NATIONAL INSTITUTE OF AEROSPACE

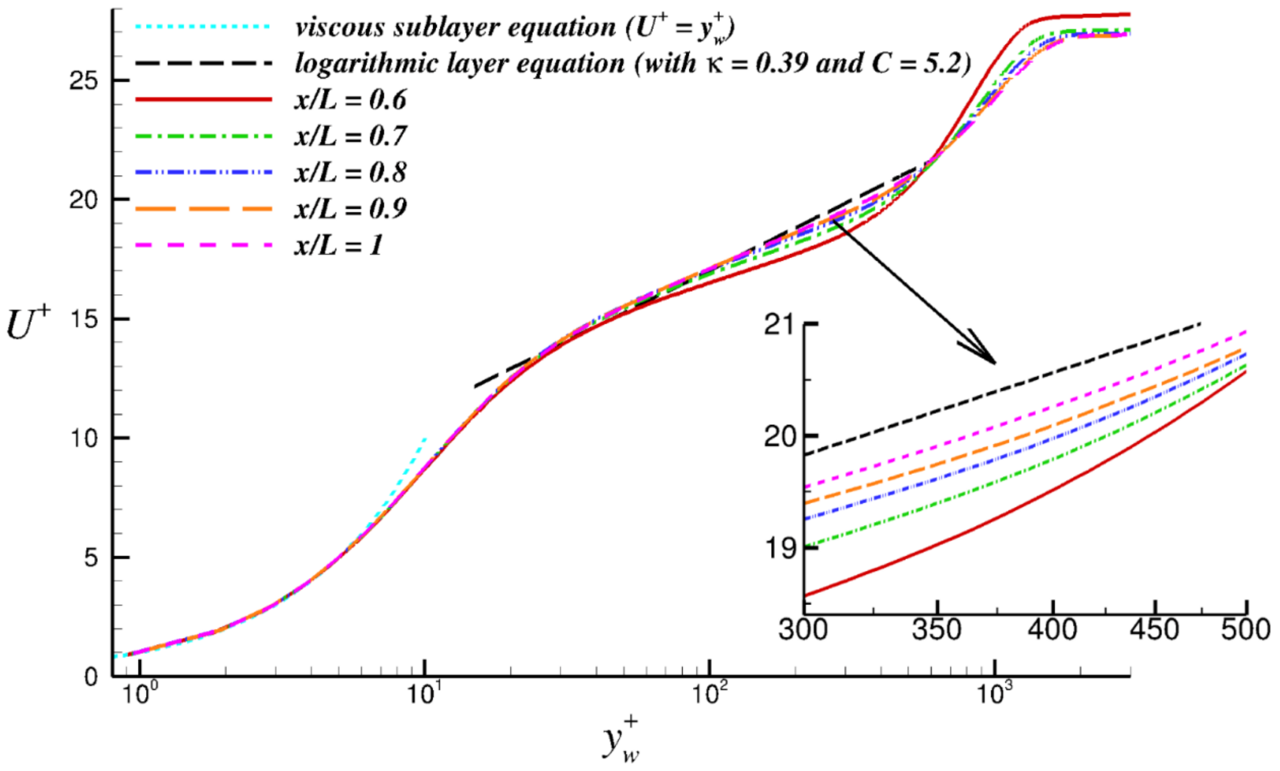
Streamwise Reynolds Stresses in Recovery Region



Evolution of the inner peak in streamwise stress is due to the formation of new internal layer in recovery region

Outer peak decays as the shear layer portion gradually disappears due to the weakly favorable pressure gradient in recovery zone

Mean Velocity Profile Development in Recovery Region



The logarithmic layer is still in adjustment as the flow nears the domain end:

- further development length is needed

Concluding Remarks

- Combination of low Reynolds number, favorable pressure gradient and convex surface curvature leads to flow relaminarization/stabilization in the accelerating region
  - Patel and Head’s threshold value of 0.018 for the acceleration parameter was found to reasonably predict the breakdown of logarithmic layer in the accelerating region
  - As expected, RANS fails to detect this phenomenon
- Present DNS predicts incipient or very weak separation in the adverse pressure gradient region
  - In comparison, RANS yields much stronger separation
- Internal layers form where the sense of streamwise pressure gradient changes at the foot, apex and tail of the speed bump

Future and Ongoing Work

- We would like to repeat the DNS at the Reynolds number of 3.5 million :
  - 150 billion grid points for the domain span of $0.1L$
  - Considering a freestream Mach number of 0.3 (to reduce run times, resources are still limited)
  - Only feasible on Summit: DNS requires 1064 Summit nodes (or 6384 GPUs)
  - 0.6 million Summit node-hours (about 23.5 days of runtime) to run $13.5\, L/U_\infty$
- Currently running a simulation at the Reynolds number of 2 million :
  - 10.2 billion grid points
  - Resolution approaches DNS in certain regions and wall-resolved LES elsewhere
  - Domain span of $0.08L$, freestream Mach number of 0.2
  - CPU code will take 50 days of run time for $10\, L/U_\infty$ using 20,000 Intel Skylake cores at NAS
  - GPU code would need 216 V100s (with 32 GB memory) for this grid, but NAS does not own that many V100s at present
  - We are unable to use our GPU code for this simulation

Acknowledgments

- This work was sponsored by the NASA Transformational Tools and Technologies Project of the Transformative Aeronautics Concepts Program under the Aeronautics Research Mission Directorate
- The calculations were made possible by the computing resources provided by the NASA High-End Computing Program through the NASA Advanced Supercomputing Division at Ames Research Center
- This work also used resources of the Oak Ridge Leadership Computing Facility (OLCF) at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy
- We thank Dr. Philippe Spalart for valuable discussions
- Drs. Michael Strelets and Prahladh Iyer provided the RANS solutions

Further Questions or Comments?

- Questions and comments can be directed to:
  - Dr. Ali Uzun: ali.uzun@nasa.gov or ali.uzun@nianet.org
  - Dr. Mujeeb Malik: m.r.malik@nasa.gov